REMARKS

Claims 1-49 were pending at the time of examination. Claims 1, 4-5, 8, 10, 15, 17, 19, 22-23, 26, 28, 33, 35, 37, 40, 44, 46 and 48 have been amended. No new matter has been added. The applicants respectfully request reconsideration based on the foregoing amendments and these remarks.

Claim Rejections – 35 U.S.C. § 112

Claims 8, 10, 26, 28 and 44 were rejected under 35 U.S.C § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which the applicant regards as the invention. In particular the language "may be" was rejected as being indefinite. The applicants have changed the wording of rejected claims to read "is operable to be" in order to remove the uncertainty about whether anything will actually happen and focus on the abilities of the recited claim elements. The applicants believe that this amendment satisfactorily addresses the rejection and respectfully request that the rejection be removed.

Claim Rejections – 35 U.S.C. § 103

Claims 1-3, 10-12, 15-21, 28-30, 33-39 and 46-49 were rejected under 35 U.S.C § 103(a) as being unpatentable over U.S. Patent No. 5,548,506 to Srinivasan (hereinafter "Srinivasan") in view of Applicant Admitted Prior Art (hereinafter AAPA).

Claims 4-9, 13-14, 22-27, 31-32 and 40-45 were rejected under 35 U.S.C § 103(a) as being unpatentable over Srinivasan in view of AAPA as applied to claims 1-3, 10-12, 15-21, 28-30, 33-39 and 46-49, and further in view of U.S. Patent No. 5,826,239 to Du et al. (hereinafter "Du").

In general, the applicants' invention pertains to an apparatus and methods for managing resources in an object-based computing system. A resource manager manages resource consumption of several resource entities. The resource entities are each capable of consuming resources and represent entities, such as applications, applets, Xlets and so on. The resources that are consumed by the resource entities include, for example, Java heaps, file descriptors, video RAM, native heaps, hardware resources, persistent storage, and so on. The resource manager tracks the availability of such resources and determines whether a resource is critically

short or reaches a particular usage level. When a resource becomes critically short or reaches a particular usage level, the resource manager selects one or more resource entities based on one or more criteria. For example, a resource entity that has the least restrictive resource usage policy or state is selected. The resource manager then requests that the selected resource entity changes its resource usage state to a more restrictive state. Of course, when resource usage reaches an acceptable level, the resource manager may also inform each resource entity (or previously selected resource entities) that they may set their resource consumption state to a less restrictive state.

Srinivasan, on the other hand, is directed to an "Auto Multi-Project Server System", which automated the tasks of project management coordination, for organizational work-group team members (Srinivasan Abstract). Activities of the automated computer based server include collating/compiling project data, flagging inconsistencies, following up with work-team members, obtain updated project tracking data, communicate project progress to work-team resources based on project priorities and generate management reports for flagging time and cost overruns and critical path information. (Srinivasan, col. 1, lines 42-50). The heart of the Auto Multi-Project Server System is a project database (10) that contains information about the various projects. The database (10) is managed by an auto project management server (20), which operates on the project data in the database (10). Users can communicate with the database (10) and the server (20) through a number of communication means (50), such as Fax, LAN, WAN, telephone network, and so on.

Turning now to the specific claim rejections, claim 1 as amended is specifically directed to "a method of managing resource usage by one or more resource entities, wherein each resource entity is configured to represent an entity that consumes one or more resources." The Examiner alleges that Srinivasan shows this in col. 3, lines 18-25 and col. 8, lines 54-57. The applicants respectfully disagree. Srinivasan neither discloses nor suggests the use of resource entities. As was described above, resource entities represent entities in an object-based computing system that consume resources, such as applications, applets, Xlets and so on. The resource entities themselves are also in an object-based computing system. All the resource entities are registered with a resource manager. In one implementation, the resource entities are objects that belong to a ResourceEntity class and interact with the resource manager through a Resource Notification List (104). The resource entities can have different states that indicate the resource consumption of their respective associated entities. The resource manager is arranged to manage the resource entities, and can perform a number of functions related to the resource

entities and general resource consumption. For example, the resource manager can query each resource entity about its state, and request that the resource entity change its state to a more restrictive state, if necessary. The interactions between the resource entities and the resource manager occur automatically in an object-based computing system, without human intervention. Thus, the resource entities and the resource manager, as claimed, are specific features directed to managing resource consumption in an object-based computing system.

Srinivasan does not show or suggest an object-based computing system. In fact, the server system (20) in Srinivasan "has been implemented in C language..." (Srinivasan, col. 6, lines 18-20), which is not an object oriented language. For this reason, Srinivasan cannot use constructs such as the resource entities and the associated resource manager discussed above. As a result, a large part of the resource management in Srinivasan is manual, and is performed by people such as program managers, project leaders, task leaders, and so on. For example, in step 4 of FIG. 4 of Srinivasan, which is directed to "Resolving Inter-project resource conflicts," the procedure determines "critical common resource usage" and compares it against the resource limits. However, as can be seen in Srinivasan, "Prior to this the Program Manager is mailed a list of projects and requested to assign a rank priority. In addition the program manager is requested to supply a list of critical resources and their usage limits. The actual usage is compared against this limit " (Srinivasan, col. 7, lines 50-55). All such a manual input can be avoided by using the resource entities and the resource manager of the applicants' invention.

The first step of claim 1 recites "in an object-based computing system, continuously determining by a resource manager with which each resource entity is registered whether a resource has reached a critical level;" That is, the resource manager continuously determines the state of the resources and whether any resource has reached a critical level, and this determination occurs in an object-based computing system. As was discussed above, Srinivasan does not describe an object-based computing system and has no resource manager. Furthermore, the cited "resource module" of Srinivasan checks for resource usage and reallocates resources if resource limits are exceeded, but this is not a continuous process, as required by claim 1. On the contrary, the process runs "at fixed intervals (example: at the end of the day)" (Srinivasan, col. 5, lines 44-50), which would clearly be insufficient for the applicants' invention.

The following two steps of claim 1 recite:

"when it is determined that a resource has reached a critical level, selecting by the resource manager a first resource entity, based on one or more criteria;" and

"when it is determined that a resource has reached a critical level, requesting by the resource manager that the selected resource entity to change its resource consumption state to a more restrictive state."

That is, if the resource manager discovers that a resource has reached a critical level, it selects a resource entity and requests that this selected resource entity changes its consumption state to a more restrictive state. As was discussed above, Srinivasan does not have a resource manager or any resource entities. Furthermore, the "resource module" of Srinivasan performs the reallocation of resources based on inter-project priorities that have been input by a Program Manager. In the applicants' invention, resources are preserved by simply requesting that the resource entities change their resource consumption to a more restrictive state.

The Examiner argues that since AAPA teaches requesting the selected resource entity to change its resource consumption state and that it would have been obvious at the time the invention was made to combine Srinivasan and AAPA to arrive at the invention. The applicants respectfully disagree. AAPA discloses requesting that applications (not resource entities) lower their resource usage. This teaching does not render claim 1 any more obvious than Srinivasan alone, since neither AAPA nor Srinivasan discloses methods in which a resource manager and resource entities are used in managing resource usage. For at least these reasons, claim 1 is neither anticipated by, nor obvious in view of the cited prior art and should be withdrawn.

Claims 2-18 all depend from claim 1 and the rejection of these claims is unsupported by the art for at least the reasons discussed above with respect to claim 1, and should be withdrawn.

Claim 19 is a *Beauregard* claim corresponding to claim 1. The rejection of claim 19 should therefore be removed for at least the reasons discussed above with respect to claim 1.

Claims 20-36 all depend from claim 19 and the rejection of these claims is unsupported by the art for at least the reasons discussed above with respect to claim 19, and should be withdrawn.

Claim 37 is a system claim reciting limitations similar to the limitations of claim 1. The rejection of claim 37 should therefore be removed for at least the reasons discussed above with respect to claim 1.

Claims 38-49 all depend from claim 37 and the rejection of these claims is unsupported by the art for at least the reasons discussed above with respect to claim 37, and should be withdrawn.

Lastly, Du, which was used in rejecting claims 4-9, 13-14, 22-27, 31-32 and 40-45, does not teach or suggest any resource manager and resource entities, as defined above, and does thus not render the applicants' invention, as defined in the rejected claims, any more obvious than the combination of AAPA and Srinivasan.

Conclusion

The applicants believe that all the rejections in the office action are now moot, and respectfully request a Notice of Allowance for this application from the Examiner. Should the Examiner believe that a telephone conference would expedite the prosecution of this application, the undersigned can be reached at the telephone number set out below.

Respectfully submitted,

BEYER WEAVER & THOMAS, LLP

Fredrik Mollborn
Reg. No. 48,587

P.O. Box 70250
Oakland, CA 94612-0250
(510) 663-1100